A Flipped Active-learning Class to Support Diverse Students in a Large Introduction to Programming Class

Prof. Laura Kay Dillon, Michigan State University

Laura Dillon is a professor and past Chair of Computer Science at Michigan State University (MSU); before joining MSU, she was a professor at the University of California, Santa Barbara. Her research centers on formal methods in software engineering, specification, and analysis of concurrent software systems. An ACM Distinguished Scientist, Laura has served on numerous editorial boards, program committees, funding panels, and advisory committees—most recently, as Vice Chair of ACM SIGSOFT and General Chair of the 38th International Conference on Software Engineering. She has participated in many CRAW and ACM mentoring events; is a founding adviser of MSU Women in Computing, the MSU ACM-W Chapter; and co-led TechKobwa, a technology camp for secondary-school teachers and female students in Rwanda, for three summers. She was awarded the ACM SIGSOFT Distinguished Service Award in 2017.

Ms. Michelle Slattery, Peak Research

Michelle Slattery has more than 30 years' experience as a professional evaluator working with diverse clients on complex project evaluations. Her business, Peak Research LLC, is celebrating 25 years of providing services to evaluate S-STEM, BPC, RED, IGERT, and I3 grants for the National Science Foundation. She is an Extension Services Consultant to the National Center for Women & Information Technology and has helped more than a dozen universities increase recruiting and retention of women in their technology programs with data driven methods and evaluation support. Four of her clients are NCWIT NEXT Award winners for the Extension Services outcomes they achieved. Michelle has an M.A. in Applied Behavioral Sciences with Human Factors specialization. Her research interests include the evaluation of STEM projects and programs, diversity studies, strategic planning, and data visualization.

A Flipped Active-Learning Class to Support Diverse Students in a Large Introduction to Programming Class

1. Introduction

Ten years of surging interest in computer science has produced a nearly 300% increase in computer science (CS) majors at Ph.D. granting institutions, as well as significant increase in demand for computing courses by non-majors [1]. While this surge may eventually help ameliorate pressing workforce needs for computing professionals [2], it is placing inordinate stress on resources and faculty of most CS departments. Measures taken to reduce this stress are often at odds with goals to increase the numbers of CS graduates and to equip graduates of other majors with the computational skills they will need to succeed in their chosen professions. Moreover, most of these measures may also have unintended negative impacts on efforts to increase diversity of CS majors.

Many CS departments are responding to surge challenges by reducing the variety and regularity of courses offered, limiting access to courses by non-majors, or raising grade requirements for pre-requisite courses and majors [1]. Unfortunately, these measures work against goals to create inclusive environments and to meet computational learning needs of non-majors. Most CS departments have also dramatically increased class sizes, and many are turning to blended or fully online courses with graduate and undergraduate assistants providing tutoring, monitoring discussion forums, or consulting services [1]. These strategies help with problems of finding enough classrooms and teachers to accommodate students but undercut many of the research-based methods proven to help retain more diverse students and women, like providing role models with Peer-Led Team Learning [3], increasing confidence and self-esteem with active learning [4], and building community and sense of belonging with collaborative learning groups and diverse instructors [5]–[8]. While both academic and non-academic factors affect persistence in all majors, non-academic factors have a stronger influence for non-majority students in engineering [9], [5].

At Michigan State University, the Department of Computer Science and Engineering (CSE) initially responded to enrollment pressures in the first two courses for majors by increasing the sizes of lectures and the numbers of lab sections. It later added an online section and eliminated lectures for all sections of both courses. The first software design course is now also taught online. As a result, a CS major might not attend a weekly course with other majors or meet a CS professor face-to-face until her junior year. This situation is not conducive to building community or a sense of belonging. CSE has also had to cut back on the number of seats for non-engineering students in many CS courses.

As an experiment to determine whether a 1-credit hour support class could use research-based methods to counter negative effects of these changes on efforts to broaden diversity, CSE developed a flipped active-learning class, called the Computational Thinking Lab (CTL). Overarching goals of the CTL include helping students excel in Intro CS[1] and increasing confidence, expertise, and community. Goals for improvement were assessed in 5 areas: overall performance and grades in the Intro CS course, understanding of material in the Intro CS course,

[1] The first programming class for CS majors, a.k.a. "the major intro course" [1] or "CS1" [20].

confidence in the Intro CS course, enjoyment of programming, and becoming a better programmer. Goals for change in attitudes and opinions were assessed in 6 areas: interest in CS careers, plans to major, intent to earn a degree in CS, feelings of belonging in Intro CS and the CTL, and perceptions of expertise. This paper describes the CTL course and documents its impact on performance in the Intro CS course as well as changes and improvements in student attitudes and experiences that have been achieved since the CTL course's inception in fall 2013.

2. Background

2.1 Institutional Context

Established in 1855, Michigan State University (MSU) is one of the nation's pioneer land-grant universities. The Department of Computer Science and Engineering (CSE) resides in the College of Engineering, one of the largest colleges with more than 6,700 undergraduate majors. CSE administers a Bachelor of Science (BS) program in Computer Science. Currently, the BS program has 1,300 majors. Although first-year students can declare an engineering major preference, most do not gain admission until their junior year. This policy impacts retention as a student is at high risk to leave until admitted. The Computing Accreditation Commission of ABET[2] accredits the CS BS program.

2.2 Intro CS

Intro CS at MSU is a 15-week, 4-credit course introducing students to solving problems through programming.[3] In this course, students design, implement, and test programing solutions to problems in engineering, mathematics and science. Weekly projects emphasize data manipulation using real-world examples,[4] which have broad appeal in showing how computing connects to students' lives and interests [8], [10]. A typical CS major takes Intro CS in her second or third semester on campus. However, a student who is not ready for Calculus I in her first semester might not take Intro CS until her fourth or fifth semester. Intro CS at MSU uses Python for programming; subsequent courses use C++.

The instructional model for Intro CS at MSU has evolved with increasing enrollments. Until fall 2013, an Intro CS student attended two weekly 75-minute lectures and a weekly 2-hour lab. At that time, CSE was offering two lecture sections of Intro CS in both the fall and spring semesters. A CS instructor delivered the lectures in a large auditorium to about 120 students each. Both then and now, a teaching assistant (TA) attends the lab section to answer questions, pair students up for pair programming [11], [12], and check that pairs complete the lab assignment. TAs also keep attendance, grade projects, monitor and respond to questions on an electronic discussion forum, and take turns staffing a "consulting room" where Intro CS students can come for help with programming.

In fall 2013, CSE added an online section to Intro CS. Students in the online section view short videos in lieu of attending lectures. They do not attend any lab section, but submit lab

[2] http://www.abet.org/about-abet/. [Accessed 1-Feb-2018].
[3] We describe only the fall and spring offerings of Intro CS since the CTL is not offered in the summer.
[4] Project archive: http://www.cse.msu.edu/~cse231/PracticeOfComputingUsingPython/ [Accessed 1-Jan-2018.]

assignments online. All sections complete the same programming projects and readings, and take the same exams.

With enrollment pressure continuing to mount, in fall 2016, CSE eliminated the lectures in Intro CS and reallocated resources to accommodate more students. Currently, CSE offers 25 lab sections of Intro CS, each with capacity of 22—24 students and supervised by a TA, and one online section, with capacity of 125 and supervised by 4—5 TAs. All Intro CS students view videos online instead of attending lectures. Students in a lab section in Intro CS attend a weekly 2-hour lab on campus, whereas students in the online section complete the lab assignment on their own and hand it in online. The consulting room is currently staffed about 25 hours per week with 1—5 TAs.

2.3 CTL support course

In spring 2010, NCWIT Extension Services helped CSE administer a Student Experience of the Major Survey to inform strategic plans for increasing the percentage of women among CS majors. It seemed like MSU was doing everything the literature recommended to recruit and retain women (pair programming, real-world examples, female TAs), but with women making up just 10% of declared majors and 7% of graduates, the department felt it could do better. Based on survey responses, CSE identified three key areas for improvement: 1) Improve the sense of community within the major; 2) Increase the use of collaborative learning and other methods that increase student engagement and support; and 3) Address introductory course methods and content to potentially attract more qualified non-majors who enroll in Intro CS. The CTL was designed with these goals in mind.

The CTL aims to help students develop strong computational reasoning skills in a challenging but supportive environment, while also building confidence and community. It borrows elements from the University of Wisconsin's Emerging Scholars Program [13] and from an Intro CS course for non-majors at University of Virginia [14]. Students sign a pledge at the start of the semester to be active learners and also to help their classmates learn. A student may elect to take the CTL only if she is concurrently taking Intro CS. She earns 1 credit for successful completion of the CTL. Her grade is based on participation.

The CTL meets weekly for 2 hours in a classroom configured for small group interactions. Students spend 80% of the class time working in small groups on exercises designed to help them understand concepts covered in Intro CS that week. In a typical class period, exercises involve several "deconstructionist" activities (e.g., reading, tracing, and debugging code), which lead up to one or more "constructionist" activities (e.g., writing code) utilizing significant scaffolding and feedback [15]. The instructor and several undergraduate peer leaders circulate through the room fielding questions and encouraging students to experiment and draw conclusions from observations. The instructor also calls students' attention to extra-curricular opportunities with MSU's student organizations and to study abroad and internship programs. Students view the Khan Academy video *How to grow your brain*[5] early in the semester to foster a growth mindset [16]. Course staff members remind students throughout the semester that

[5] https://www.youtube.com/user/khanacademy/search?query=How+to+grow+your+brain [Accessed: 1-Feb-2018.]

struggling to understand concepts is not just to be expected, but also necessary to learn. The peer leaders are students who attended the CTL in the previous year.

3. Methods

3.1 Intro CS course

Students: New enrollments in Computer Science increased by approximately 88% from fall 2013 to spring 2017 at MSU. During this period, a total of 3,102 students took the Intro CS course. Female students made up 20.3% of the Intro CS course during the CTL era. Student race included 61% white, 17% international, 10% Asian, 5% black/African American, and 3% reported two or more races. Three percent of students reported Hispanic ethnicity. Just 45% of students in the Intro CS course came from engineering majors (including CS). Nearly 200 different majors were represented in the Intro CS course.

3.2 CTL support course

Students: The CTL course is promoted as helping students develop strong computational reasoning skills in a challenging but supportive environment. Advisors recommend it to Intro CS students who have limited prior programming experience or who are women or minority students. All students who enroll in Intro CS are sent an email describing the CTL course. Over 9 academic semesters, 268 students participated in the CTL. Approximately 9% of Intro CS students took the CTL support class. Female students accounted for 36.9% of CTL participants, compared to 20% in the Intro CS course. Race was also slightly more diverse in the CTL, with white students making up 55% of the CTL class compared to 61% in Intro CS. Black/African American students made up 8% of the CTL, compared to 5% of Intro CS. Asian students made up 16% of the CTL, compared to 10% of Intro CS. International students were less well represented in the CTL at 13% compared to 17% in the Intro course. Majors represented in the CTL were predominantly from engineering and computer science (67%), but 22 additional majors were represented by a single student. The non-engineering majors included everything from Russian to Criminal Justice, Music Education, and Advertising, demonstrating the growing recognition of the importance of computer science to every discipline.

To determine whether the CTL attracted a different type of student who might perform better in the Intro course, regardless of the CTL support, grade point averages (GPAs) of the two populations of students were compared. Mean GPAs were nearly identical for Intro CS and CTL populations, at 3.182 and 3.177, respectively. Registrar data were also obtained on number of student attempts to pass the Intro CS course to determine whether differences in persistence between the Intro CS and CTL populations might explain other differences. The number of attempts were nearly identical with Intro CS students taking the Intro CS course an average 1.06 times, compared to the CTL average of 1.05 times.

3.3 Data sources and analyses

CTL surveys: Computing and engineering attitude surveys were reviewed [17], [18] to determine whether validated instruments exist that could succinctly capture the unique constructs of interest in the CTL support class without significantly increasing the evaluation burden to students. In addition to items about computing attitudes, expertise, intent, and impact, researchers hoped to learn more about student programming experiences, tutoring hours by subject, goals for participation, suggestions for improvement of the CTL, and suggestions for improvement of the College of Engineering. The authors never intended to develop and validate a new scale for the CTL class. However, to ensure that the items used addressed the goals of the project, internal consistency of the constructs was reviewed using SPSS to calculate Cronbach's Alpha. Cronbach's Alpha of .839 showed that the items used had good internal consistency. The authors also explored internal consistency by examining individual item responses to confirm that they agree with each other. Ultimately, eleven items were developed for assessing computing attitudes, expertise, intent, and impact of the CTL class. Future study of the items with factor analysis or principal components analysis should be undertaken to ensure unidimensionality.

A total of 165 CTL students completed email surveys over 5 total semesters. Data from 4 additional semesters could not be aggregated and compared due to changes in item wording, such as the addition of a neutral response option.[6] A Retrospective Pre-Post Test (RPT) methodology [19] was used on six items to reduce rater bias and to increase evaluation efficiency for the classroom. Surveys asked participants to provide ratings of agreement pre-and-post taking the CTL course on statements related to their interest in a career in computer science, plans to major in computer science, plans to earn a degree in computer science, and feelings of belonging in both Intro CS and the CTL. Students in the CTL were also asked to provide ratings of expertise in computer science both pre-and-post CTL. As all were concurrently taking Intro CS, ratings on these surveys reflect impact of the combination of Intro CS with the CTL, not of the CTL by itself. These items were analyzed with 2-tailed paired samples t-tests (95% CI) to determine the significance of rating differences before and after the CTL support class. Ratings of agreement on statements related to goals of the CTL course were also obtained. Mean ratings of agreement and descriptive statistics are reported. We report standard error for mean ratings for all Intro CS students, but not for ratings of CTL-only students because of the small sample size.

Grades: Institutional research data were provided for the grades of all Intro CS students during the CTL era (Fall 2013 – Spring 2017) when the support class was offered, for a total of 3,102 students. Student grades ("A" through "F") were provided, along with Drop/Withdrawal status, and CTL participation status. A total of 268 students completed both the Intro CS and CTL courses concurrently. In total, 2,834 Intro CS students did not take the CTL support course. Mean grades were compared for the Intro CS and CTL support class samples using independent samples t-tests (95% CI). Levene's test for homogeneity of variances was used. Effect size was calculated with Cohen's *d* to determine the size of the difference between the means.

[6] We opted to not include a neutral response to reduce the "satisficing" tendency in attitude surveys [21], [22]. For accuracy, however, respondents were not required to respond to each item in order to proceed.

NCWIT tracking tool data**:** Data on new enrollments, declared majors and graduation rates by gender and year for computer science majors were available for the 2010 through 2016 academic years. Women's share of new enrollments, declared majors, attrition, and graduation rates were tracked. Male and female data on attrition rates over time were also compared. These data were entered into the NCWIT Tracking Tool[7] for comparison over time, by gender, and against national averages for graduation rates. Data were explored three years prior to the start of the CTL support class being offered to establish baselines (2010-2012). Data were explored after the launch of the CTL support class to track potential impact and improvements (2013-2016). Computing Research Association (CRA) Taulbee survey data[8] were available through 2015 for comparison. The Taulbee Survey is the principal source of information on graduation trends in computer science education. These graduation data show the percentage of Bachelor's degrees awarded each year at institutions that have Ph.D. programs in computer science. Taulbee data provide a benchmark for comparison of graduation trends at universities across the nation. The National Center for Education Statistics-Integrated Postsecondary Education Data System or IPEDS data provide the percentage of Bachelor's degrees awarded each year at public and private, not for profit, institutions with 4-year or above degrees. Data are aggregated from Classification of Instructional Programs (CIP) Codes. The computer science (11.07) CIP code was selected for comparison of graduation trends.

Focus groups: Focus Groups were conducted via Zoom Conferencing with two different groups of students during spring semester 2017. The first group consisted of 4 freshmen enrolled in Intro CS and the CTL during spring semester 2017. The second group consisted of 5 seniors and 1 sophomore who reflected on the experiences they had as underclassmen, 1-3 years earlier, in the CTL and Intro CS. Participants were recruited via email sent to all students who were enrolled at MSU in spring 2017 and either were taking or had previously taken the CTL. Participation was voluntary and confidential. Participants received a $25 gift card as incentive. They participated from the location of their choice via Zoom Conferencing. Discussions lasted approximately one hour and were recorded and transcribed for analysis. Participants were asked about the impact of the CTL experience on their retention in the major, their development as computer scientists, and their confidence, sense of belonging, and performance in the Intro CS course.

4. Results

4.1 Grades in the Intro CS course.

The non-CTL group of students (N=2689) had mean grades of 2.53 (SD = 1.304) in the Intro CS course. By comparison, the CTL group of students (N = 268) had numerically higher mean grades of 2.76 (SD = 1.188) in the Intro CS course. To test the hypothesis that the non-CTL and CTL students had grades that were statistically different at 95% CI, an independent samples t-test was performed. In Table 1, the non-CTL and CTL distributions were sufficiently normal for the comparison. The assumption of homogeneity of variances was tested and satisfied with Levene's F test, $F(2955) = 5.35$, $p = .021$. The independent samples t-test was associated with a statistically significant effect, $t(2955) = 2.66$, $p = .008$. The CTL students were associated with

[7] https://trackingtool.ncwit.org/ [Accessed 30-Mar-2018.]
[8] https://cra.org/resources/taulbee-survey/ [Accessed 30-Mar-2018.]

grades in the Intro CS course that were statistically significantly higher than students not in the CTL support class. Cohen's *d* was estimated at .184, which is a small effect size *r* (.092), according to Cohen (1992). Cohen's *d* was calculated because the non-CTL sample in the Intro CS class was very large, most likely inflating the power of the t-test. While the grades of the two groups were significantly different ($p<.05$), the size of the difference in grades was small.

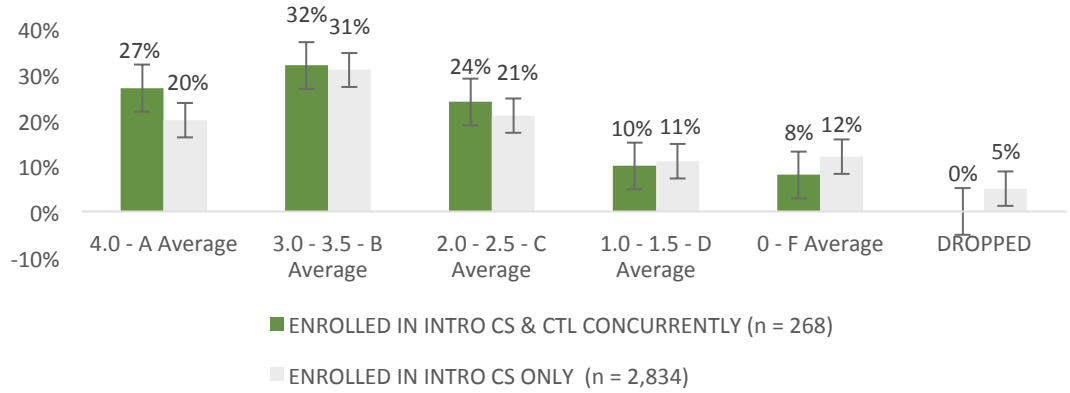
To explore further the variability of data of the Intro CS population grades and the CTL class sample grades, confidence intervals (95%) were calculated and are presented in the grades chart below as error bars. The error bars show the true grades for each group 95% of the time.

Table 1. Intro CS Student Grade Statistics

	N	MEAN	STD DEV	STD ERROR MEAN
CTL STUDENTS	268	2.76	1.188	.073
NON-CTL STUDENTS	2689	2.53	1.304	.025

None of the CTL students dropped or withdrew. Students who took the Intro CS and CTL courses concurrently were more likely to earn the highest grade of "A" and less likely to earn the lowest grade of "F" or Drop/Withdraw from the Intro course. Students enrolled in the CTL and Intro CS courses concurrently were more likely to have a passing grade (A-C average) in the Intro course (83%) than students without the CTL support class (72%). Students enrolled in the CTL and Intro CS course concurrently were less likely to fail Intro CS (18%) than students without the CTL course (23%). Improvements in grades are important because women are more likely to switch out of male-dominated STEM majors in response to poor performance [18].

Chart 1. Comparison of Grades in the Intro CS Course with & without the CTL Support Course



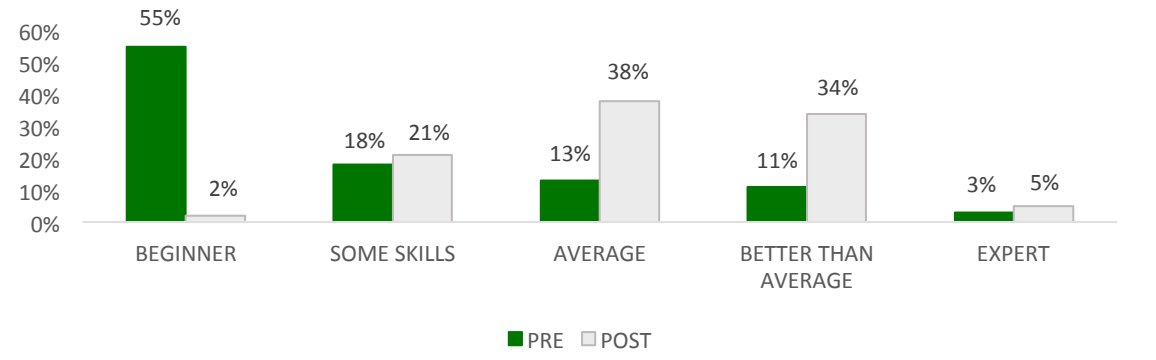
4.2 Computing Expertise.

In the pre-and-post retrospective surveys, CTL students rated their level of computing expertise on a 5-point scale from "Beginner" to "Expert." A paired-samples t-test was conducted to compare student ratings of computing expertise before and after the CTL. There was a significant difference in student ratings of computing expertise before the CTL class ($M = 1.79$,

SD = 1.080) and after the CTL class (*M* = 3.20, *SD* = .880); *t*(124) = -17.173, p = .000. These results suggest that the combination of the CTL and Intro CS had a significant positive effect on student ratings of computing expertise.

The CTL was created for and marketed to students who lacked prior computing experience and confidence, as well as to women and minority students to help them find community and reduce stereotype threat. Self-ratings of expertise pre-and-post revealed that 14% of CTL participants considered their expertise was "Better than Average" or "Expert" before taking the course. During focus groups, several students with significant high school preparation for computer science revealed that they took the CTL course because it could be substituted for a required 1-credit course on computing professionalism and ethics and industry practice. Fifty-five percent of CTL survey respondents rated their expertise before the Intro CS course as "Beginner" level. Just 2% of CTL students felt that they were still at a "Beginner" level by the end of the course. The CTL and Intro CS courses effectively brought nearly three quarters of participants up to an average level of expertise from pre-to-post course. "Expert" level ratings did not change much from pre-to-post, with just a 2% increase. A known advantage of the RPT approach is that learners "know what they don't know" when they provide their pre-ratings. The focus group results provide more insights into the improvements in computational thinking that took place for "Beginners" and "Experts" alike.

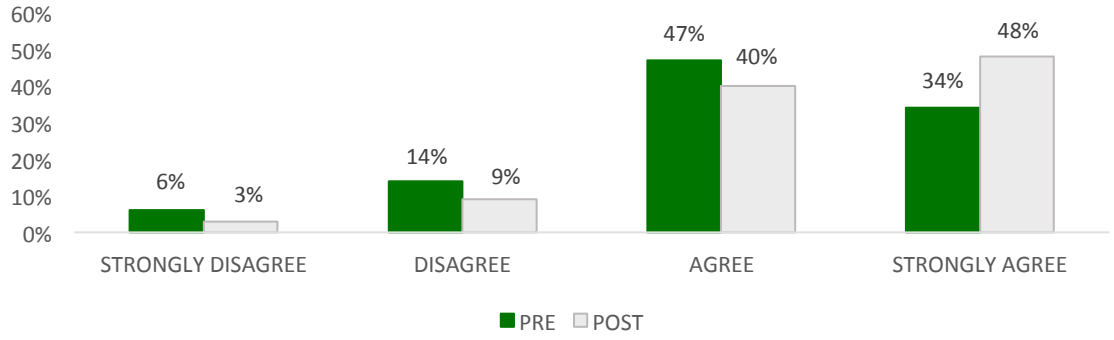
Chart 2. Student Ratings of Computing Expertise Pre-and-Post CTL Support Course



4.3 Interest in a career in computer science.

Students in the CTL course were asked to rate their agreement with this statement: "I am interested in a career in computing." Both pre-and-post ratings were provided with the retrospective method (RPT) described previously. Survey responses were analyzed with 2-tailed paired samples t-tests (95% CI). There was a significant difference in student ratings of interest in a career in CS before the CTL class (*M* = 3.09, *SD* = .835) and after the CTL class (*M* = 3.33, *SD* = .767); *t*(159) = -4.732, p = .000. These results suggest that the CTL and Intro CS courses had a significant positive effect on student interest in careers in CS. Though ratings of agreement were high both pre-and-post CTL, student ratings of agreement were significantly higher after the CTL. The CTL and Intro CS courses were effective in increasing student interest in careers in computer science.

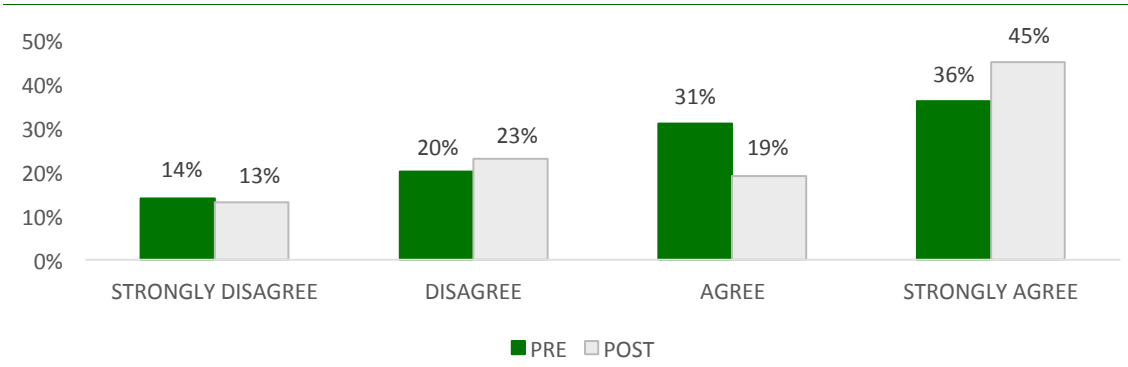
Chart 3. Student Ratings of Agreement on Career Interest in CS Pre-and-Post CTL



4.4 Intent to earn a degree in computer science.

Students in the CTL class were asked to rate their agreement with this statement both pre-and-post CTL: "I will earn my degree in computer science." Ratings were provided using the retrospective method (RPT). Survey responses were analyzed with 2-tailed paired samples t-tests (95% CI) to compare the intent of students to earn degrees in computer science pre-and-post CTL class. There was a significant difference in student ratings of intent to earn the CS degree before the CTL class ($M = 2.88$, $SD = 1.048$) and after the CTL class ($M = 2.97$, $SD = 1.096$); $t(123) = -1.999$, $p = .048$. These results suggest that the CTL and Intro CS had a significant effect on student intent to earn degrees in CS. More than a third of CTL students disagreed with the statement both before and after the CTL. They did not intend to earn a degree in CS before the CTL and the CTL experience did not change that. "Strongly Agree" ratings increased from 36% to 45% from pre-to-post, however, showing that the courses made some students even more committed to earn a degree in CS.

Chart 4. Student Ratings of Agreement on Intent to Earn the CS Degree Pre-and-Post CTL

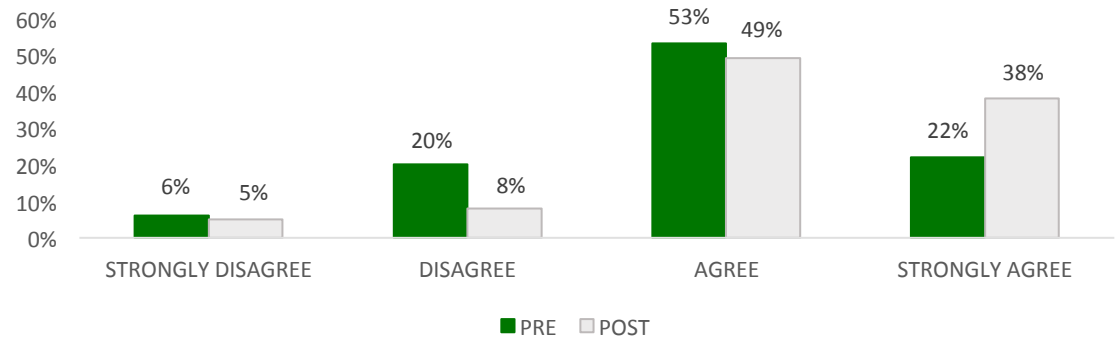


4.5 Belonging in the CTL and the Intro CS courses.

Students in the CTL were asked to rate their agreement about feelings of belonging in both Intro CS and the CTL. Both pre-and-post ratings were provided using the retrospective method

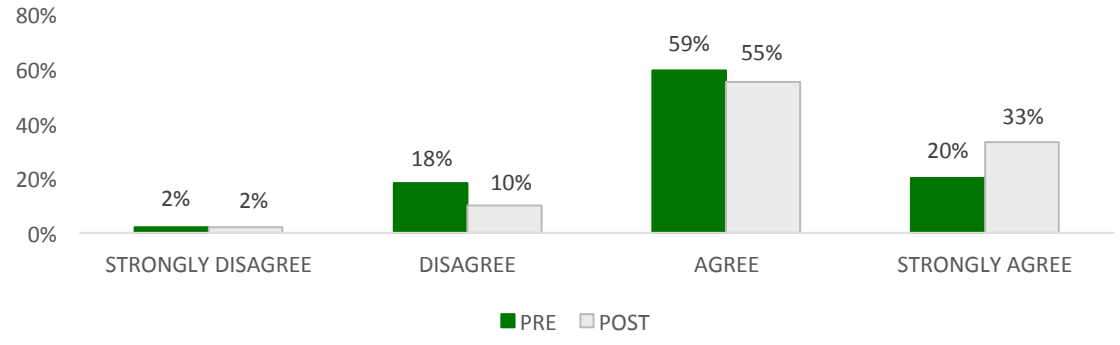
(RPT). For the first item, students provided ratings of agreement with the statement, "I feel like I belong in the Intro CS course." Survey responses were analyzed with 2-tailed paired samples t-tests (95% CI) to compare CTL students' feelings of belonging in the Intro CS course pre-and-post taking the CTL. There was a significant difference in student feelings of belonging in the Intro CS course before the CTL course ($M = 2.91$, $SD = .799$) and after ($M = 3.20$, $SD = .794$); $t(158) = -4.439$, $p = .000$. Taking the CTL course concurrently with Intro CS had a significant positive effect on student feelings of belonging in the Intro CS class. This finding suggests that augmenting an Intro CS course that has become increasingly isolating due to the surge in enrollments and conversion to an online platform with 1-credit hour flipped active-learning course utilizing peer leaders can help build community.

Chart 5. Student Ratings of Belonging in the Intro CS Course Pre-and-Post CTL



Students were also asked to rate their agreement with the statement, "I feel like I belong in the CTL course." Survey responses were analyzed with 2-tailed paired samples t-tests (95% CI) to compare the feelings of belonging in the CTL course pre-and-post. There was a significant difference in student feelings of belonging in the CTL course before the CTL ($M = 2.97$, $SD = .695$) and after ($M = 3.19$, $SD = .680$); $t(124) = -3.048$, $p = .003$. These results suggest that the CTL course had a significant positive effect on student feelings of belonging in the course. The methods used in the course were effective at building community.

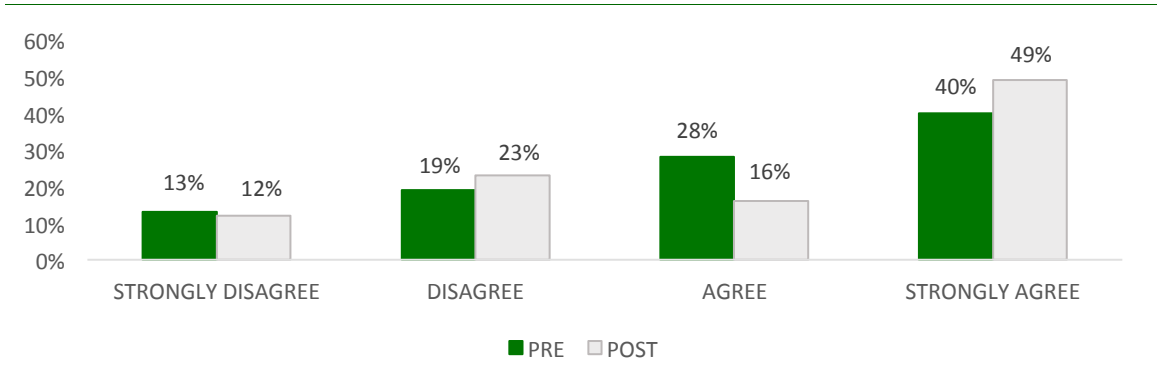
Chart 6. Student Ratings of Belonging in the CTL Pre-and-Post



4.6 Plans to major in computer science.

Students in the CTL were asked to provide ratings of agreement with this statement: "I plan to major in Computer Science." Both pre-and-post ratings were provided using the retrospective method (RPT). Survey responses were analyzed with 2-tailed paired samples t-tests (95% CI) to compare the plans of students to major in CS pre-and-post CTL course. There was no significant difference in student ratings of plans to major in CS before the CTL course ($M = 2.93$, $SD = 1.053$) and after ($M = 3.02$, $SD = 1.096$); $t(159) = -1.553$, $p = .122$. These results suggest that the CTL did not have a significant effect on student plans to major in CS. Thirty-two percent of CTL students "Strongly Disagreed" or "Disagreed" with the statement before the CTL course, compared to thirty-five percent after. These results may demonstrate increased interest in computer science courses without the intent to major. In fact, 84% of students in the CTL reported majors in the College of Engineering, with 16% taking the Intro CS course in other majors. Students may simply take the CTL support class to increase success in the Intro CS course. The CTL course did not significantly change plans to major.

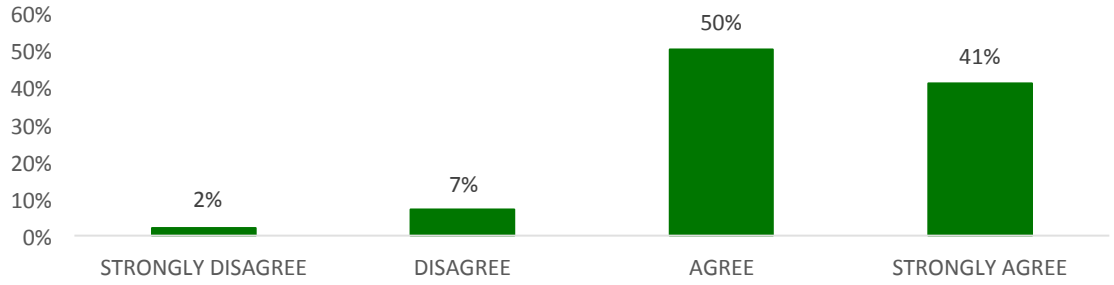
Chart 7. Student Ratings of Plans to Major in Computer Science Pre-and-Post CTL



4.7 Confidence.

After the CTL course, students were asked if they agreed with this statement: "The CTL class improved my confidence in the Intro CS course." Mean ratings of agreement were 3.32 on a 4-point scale, "Strongly Disagree" to "Strongly Agree." Students reported the CTL support course was effective in making them feel more confident about the Intro CS course, with 91% expressing positive agreement.

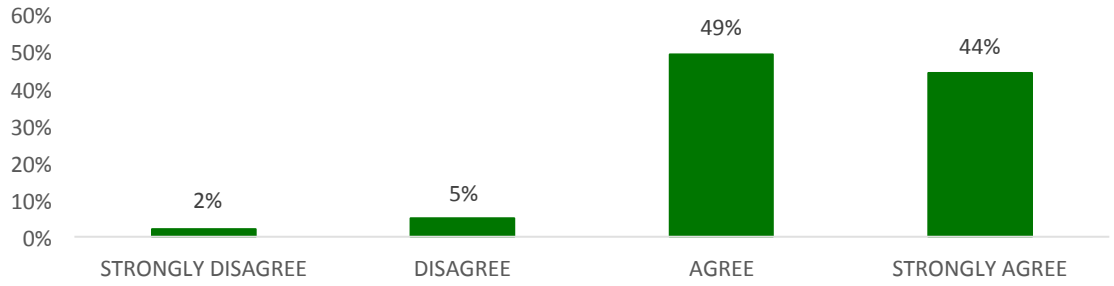
Chart 8. CTL Class Improved Confidence in the Intro CS Course



4.8 Understanding.

After the CTL course, students were asked if they agreed with this statement: "The CTL class improved my understanding of the material covered in the Intro CS course." Mean ratings of agreement were 3.37 on a 4-point scale from "Strongly Disagree" to "Strongly Agree." Students reported the CTL was effective in helping them understand the material covered in the Intro CS course, with 93% expressing positive agreement.

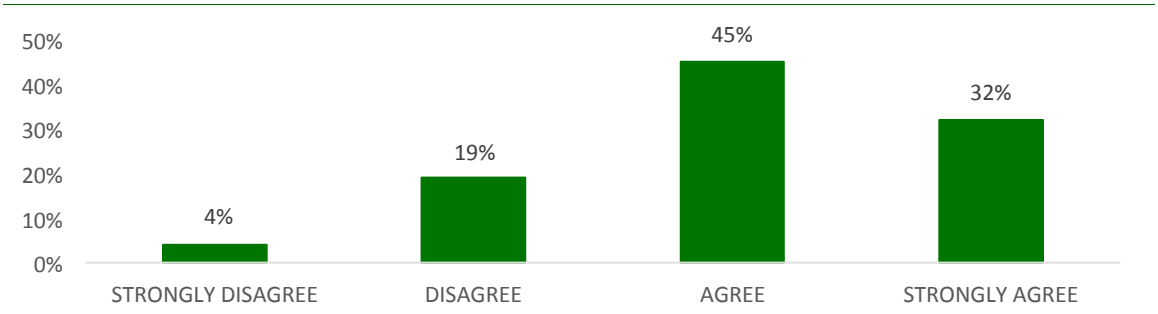
Chart 9. CTL Class Improved Understanding of the Material in the Intro CS Course



4.9 Enjoyment of programming.

After the CTL course, students were asked if they agreed with this statement: "The CTL class increased my enjoyment of programming." Mean ratings of agreement were 3.06 on a 4-point scale from "Strongly Disagree" to "Strongly Agree." The CTL support course was somewhat effective in increasing student enjoyment of programming, with 77% expressing positive agreement. Nearly a quarter of students, however, did not find programming to be more enjoyable as a result of the CTL.

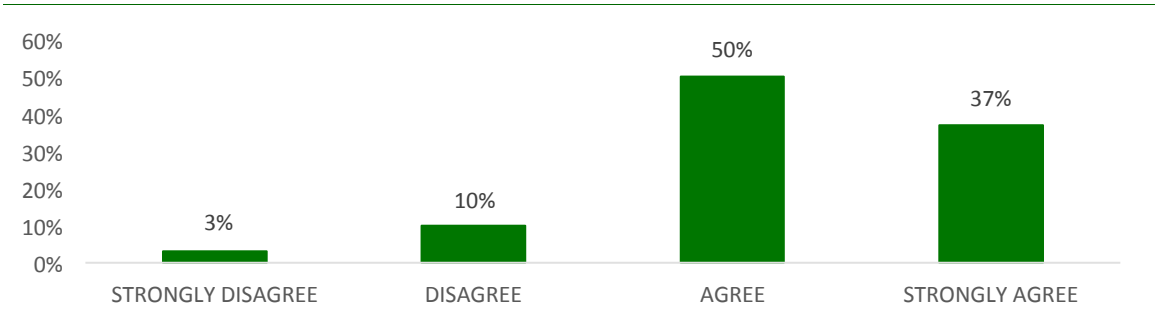
Chart 10. CTL Class Increased Enjoyment of Programming



4.10 Better programmers.

After the CTL course, students were asked if they agreed with this statement: "The CTL class made me a better programmer." Mean ratings of agreement were 3.21 on a 4-point scale from "Strongly Disagree" to "Strongly Agree." Results suggest the CTL was effective in increasing student perceptions of their skills as programmers, with 87% expressing positive agreement. Just 13% of students felt that the CTL had not made them better programmers.

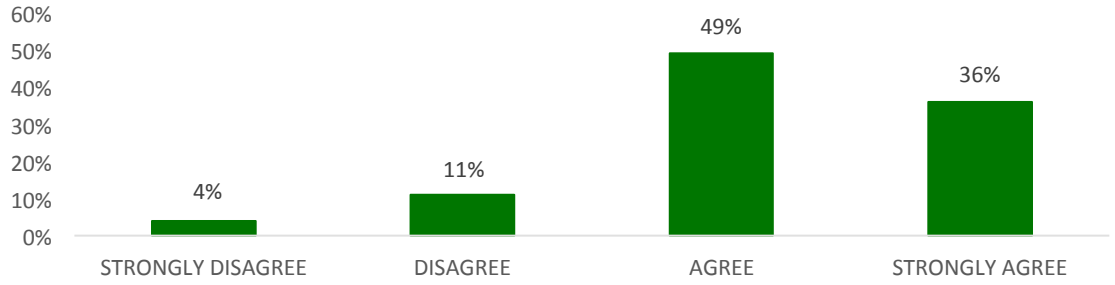
Chart 11. CTL Class Created Better Programmers



4.11 Performance and grades.

After the CTL course, students were asked if they agreed with this statement: "The CTL class improved my performance and grade in the Intro course." Mean ratings of agreement were 3.2 on a 4-point scale from "Strongly Disagree" to "Strongly Agree." Students reported the CTL support course was effective in helping them improve their performance and their grades in the Intro CS course, with 85% expressing positive agreement. Just 15% of students did not agree.

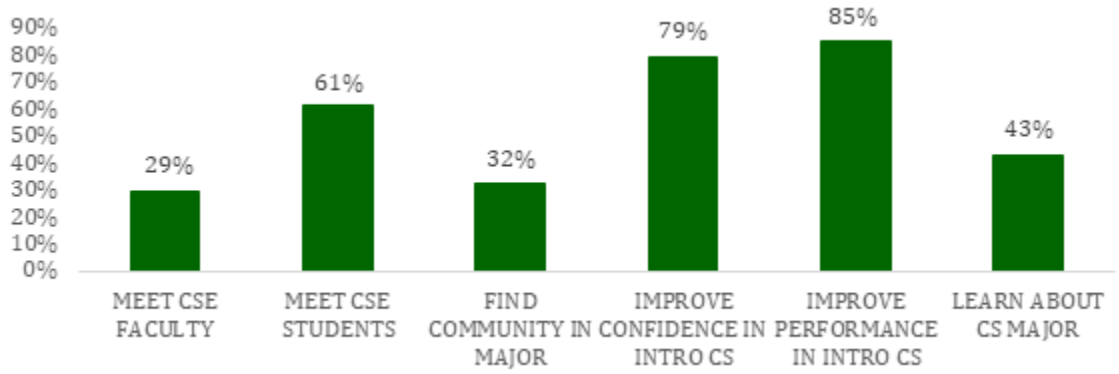
Chart 12. CTL Class Improved Performance & Grades in Intro CS



4.12 Goals for taking the CTL.

Students were asked to select from a list of goals to help researchers understand their motivation for taking the extra CTL support course concurrently with Intro CS. Students could choose all responses that applied to them. The top 2 reasons cited were to improve performance and confidence in the Intro CS course. More than half of students also cited meeting other students in the major as a goal for CTL participation, perhaps reflecting how little community is created when students participate mostly online. It is not surprising that 43% said they want to learn more about the CS major, considering that so many different majors were represented in both the Intro CS and CTL courses.

Chart 13. Goals for Taking the CTL Support Class Concurrently with Intro CS

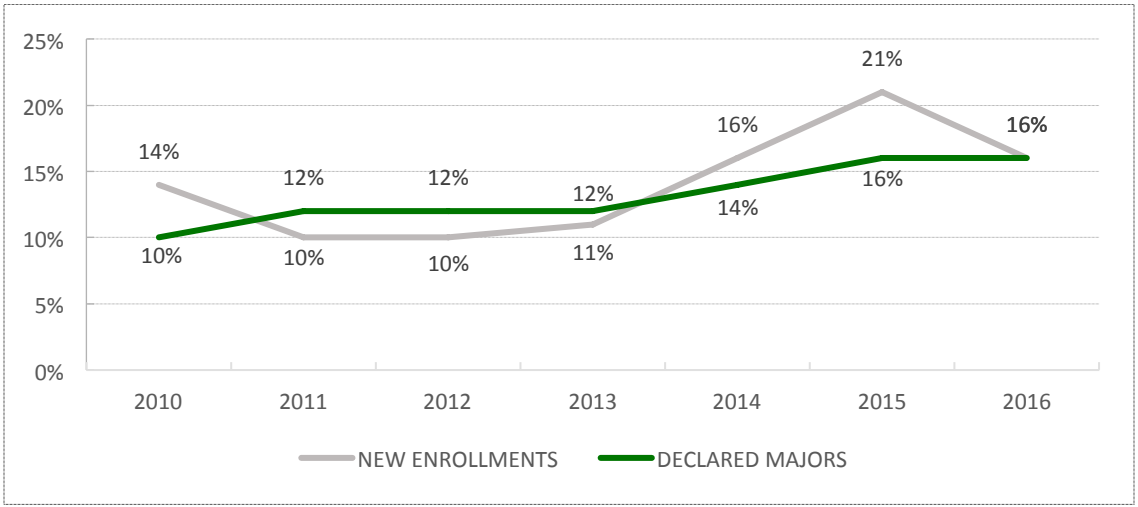


4.13 NCWIT tracking tool.

New enrollments in computer science by women at MSU ranged from 10% to 21% over seven academic years, but increases were not consistent over time. The highest rate of new enrollments by women was achieved in 2015, while 2011 and 2012 were low points. Women's share of declared CS majors ranged from 10% to 16%. Gradual, but steady increases occurred over time. Women enrolled in computer science at higher rates from 2014 to 2016, but there is a gap in new enrollments and declared majors that may indicate need for supports that build community and increase confidence by developing computational thinking skills. Ideally, female students who are interested in CS will declare the major, persist, and graduate. The CTL was implemented at a critical time, in 2013, when female interest in the major was starting to

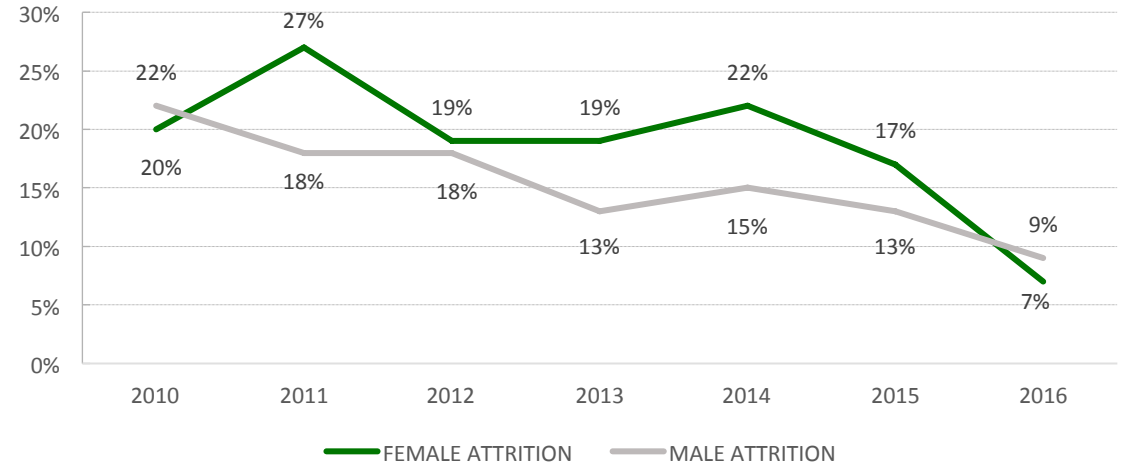
grow. Although CTL outcomes cannot be directly linked to these trends in the enrollment data, nearly half of all CTL participants have been female.

Chart 14. Female CS Enrollments & Declared Majors



Attrition rates for both male and female students at MSU have improved from 2010 to 2016. Attrition for women ranged from 7% to 27%. Attrition for women was at its highest point in 2011 and its lowest point in 2016. In 2016, gender parity with respect to retention was achieved and exceeded, with 93% of women retained compared to 91% of men in the CS major.

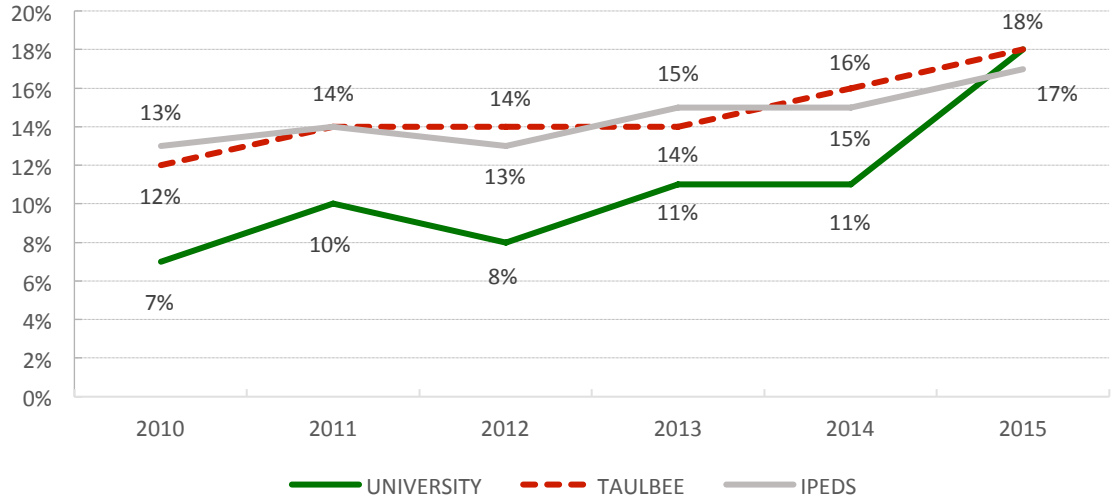
Chart 15. Male & Female Attrition Rates in the CS Major



Graduation rates for women in computer science at MSU have increased slowly, but steadily over time, with the exception of a slight drop in 2012, possibly as a result of the recession in 2008. Graduation rates for women in CS increased from 7% in 2010 to 18% in 2015. Graduation rates for women are finally keeping pace with Taulbee trends in 2015, and narrowly

outpacing IPEDS trends in 2015. When 2017 data are published, the first class of female CTL participants will be on track to graduate.

Chart 16. Comparative Graduation Trends for Females in Computer Science



Taulbee data from 2010 through 2015 show a gradual but steady increase in female computer science graduation rates from 12% to 18%. From 2010 through 2015, IPEDS graduation rates for women in computer science ranged from 13% to 17%. Graduation rates for women have increased gradually over time, with the exception of a slight drop in 2012, despite rapid growth in CS enrollments during the same period.

4.14 Focus groups. Reflections on computational thinking.

Discussions with seniors and freshmen provided an interesting comparison of the impact of the CTL from both short-term and long-term perspectives. In addition, freshmen experienced the Intro CS course as an online class, while the seniors took the Intro CS course as a traditional lecture. The CTL course was equally valuable to both novice and experienced programmers. Novices felt the class created community that helped them persist and be retained in the major. Novice CTL students reported that confidence increased and they felt it "leveled the playing field." Experienced students reported that early in the semester, the CTL helped them keep from falling behind if they did not have time to view all assigned videos or complete all readings. Later in the semester, it helped them learn about computational thinking, creating a foundation that helped them throughout their studies at MSU. Experienced programmers reported that they were surprised and overwhelmed by the difficulty of projects toward the end of the Intro CS course, so it was valuable for them as well. Regarding the Intro CS course, one participant said:

> "I think especially since it's online now, I feel like they should try to do a better job of pairing people up and letting them know who else is in computer science and just try to make friends within your major … but it's all online and you miss the face-to-face connections that you can make, either in a lecture or in a lab when you're just turning in something to hand in."

Because students at MSU don't typically gain admission to the College of Engineering until their 5th semester of enrollment, there is a long period of time during which they can discover other majors, lose confidence, or feel disconnected from the CS major. The CTL builds that community, increases confidence, and provides opportunity to collaborate with other people who share the same major. In the words of one focus group participant:

> "… my very first perception I had from the CTL was that, we're here, we're going to help each other. We want everyone to succeed. It was never like one student over another, against another. It was a collaborative environment, so I think that really sent me out on a positive note in my perception of the department."

Students had many suggestions for improving the Intro CS course. In many ways, their suggestions sounded like they would teach the Intro CS course in the same way that they were taught in the CTL: learning debugging and computational thinking, working together, building community and confidence. They want basics to be taught and resources to be shared. Senior students, especially, shared many examples of ways in which they were better programmers because of the early foundational learning in the CTL support course. One example:

> "...this thought just kind of struck me recently...in order to be a more successful programmer...computational thinking, like that is the core of it, and some people maybe develop it too late...this semester I've been helping a sophomore who's currently in 231 (Intro CS), but not the CTL and I can just tell... Breaking the problem down and approaching, like tackling the small problems aren't there for him and he doesn't see like, how can I get from this massive problem that seems really daunting to breaking it down to these smaller, simple ones that I know I can solve, and I think the CTL helps bridge that gap."

Appendix A contains other relevant comments made by focus group participants.

5. Discussion

We examined the impact of a 1-credit hour support course, called the Computational Thinking Lab, on student grades, attitudes, and experiences in the Intro CS course it supported. Our findings suggest that it is a promising, low cost, high impact method for helping students achieve milestones proven to impact their recruiting and retention at a time when resources within departments are quite challenged.

The CTL course evolved from student feedback collected with the NCWIT Student Experience of the Major survey as well as observations of an experienced CS teacher who understood the common errors that undermine beginner students in CS. Research identified the most important areas for improvement and the literature helped identify methods and strategies to increase engagement and learning. In some ways, the CTL approach seemed too basic and too simple to be effective. How could a 1-credit hour support class supported by one instructor and a few close peers create students who are confident, competent, and more resilient?

In terms of retaining students, we found that taking the CTL alongside Intro CS significantly increased intention to earn a degree in CS. Intention to earn a degree gets at commitment. A student may declare a major but not be certain of ability to complete the major, or even interest in completing the major. Increasing intention to complete the degree is important for resilience.

The research literature often cites the differences in perceptions of expertise between male and female students in computer science. Female students achieve higher grades but leave the major because of perceptions of not being good at programming. The CTL helped to normalize perceptions of computing expertise by providing diverse students opportunities to work with others like them. Student ratings of expertise increased significantly after the CTL.

Taking the CTL with Intro CS was effective at building community and significantly increased feelings of belonging in both the CTL and the Intro CS courses. About a third of students in the CTL cited finding community in the major as a goal for taking the CTL. Even more CTL students (61%) cited meeting other CS students as a motivating factor for taking it. These responses demonstrate the increased need for connection, especially in large, anonymous lecture halls or online platforms. It is interesting that the community created in the CTL had far-reaching impact, making the larger Intro CS course feel more inclusive.

Students were motivated to take the CTL to improve performance (85%) and confidence (79%) in the Intro CS class. The CTL achieved these goals. Students reported better understanding of the material covered in the Intro course, increased confidence in the Intro course, and perceptions of being better programmers. Although mean grades in the Intro CS course were slightly higher for students in the CTL, a more important measure may be student ratings of performance and grades in the Intro course due to the CTL. Eighty-seven percent of CTL students agreed that the CTL course improved their performance and grades in the Intro course. Overall, the CTL helped students perform better in the Intro CS course, feel like they could do the work, and feel supported in their efforts.

In terms of recruiting students, for example from other majors who see the importance of CS to enhance other degrees, we found that taking the CTL with Intro CS was not effective at changing student plans to major in CS and it did not make programming more enjoyable for all students, but it did significantly increase interest in CS careers for all students.

Focus Groups explored the short-term and long-term impact of the CTL on students who were currently enrolled in the CTL and students (mostly seniors) who had taken the CTL in the past. These discussions made it clear that students build community in the CTL that impacts how they feel about the Intro CS course, the major, and their ability to persist. They develop computational thinking skills with the active learning methods used in the CTL that increase their interest and commitment to the major as well as their perceptions of expertise. All of these skills and attitudes contribute to their resilience and retention in the major.

Retention has been slowly and steadily improving for both women and men in the CS major during the CTL era at MSU. Despite surveys of CTL students that show significant impact on indicators of retention, like feelings of belonging and perceptions of expertise, the CTL class touched just 9% of all Intro CS students during this time and can't be directly credited with improvements in the overall retention statistics. New enrollments for women in CS have been gradually increasing since the start of the CTL era at MSU, but there are gaps in declared majors that show that there is a need for supports to retain them through graduation.

The CTL followed the Intro CS curriculum but provided students with opportunities to practice active learning with more diverse peers and peer leaders. The CTL provided a safe space to

learn by doing, which in turn created a sense of community and belonging. This approach could be easily replicated with any program in the nation, regardless of platform. It worked well with lecture-based Intro CS courses as well as online classes at MSU. The CS Department at a large public university in the West replicated the CTL experiment as part of an NCWIT Mini Grant and achieved similar outcomes, demonstrating that the approach has broad application and reach. The CTL has been institutionalized at MSU, but it has not yet expanded the reach to accommodate more than 10% of Intro CS students each semester. Conversely, the university that piloted the CTL approach in the West is attempting to change the Intro CS course so that the CTL is integrated into every section.

Ideally, the Computational Thinking Lab should be available to all students. Its impact on the attitudes, experiences, and performance of participants in the Intro CS course was substantial. It is also interesting to consider the impact of a single 1-credit hour support course taken by less than 10% of all students on the behemoth CS major of more than 1,300 students. CTL connections and computational skills may have had a cascading impact on the larger class by introducing more students with confidence and competence to become role models. Further study is merited.

Our findings suggest that despite the small overall number of students in the Intro CS course who were able to participate concurrently in the CTL over the 5 semesters studied, the impact of the CTL on their Intro CS course experiences was encouraging. In an age of surging enrollments, when CS departments may be forced to implement survival measures that negatively impact the recruitment and retention of underrepresented students, the CTL provided a low-cost and limited time investment option for students that nurtured connections, skills, and experiences to sustain them throughout their education and possibly their careers.

References

[1] Computing Research Association, "Generation CS: Computer Science Undergraduate Enrollments Surge Since 2006," 2017.

[2] Committee on Workforce Needs in Information Technology and et. al., "Building a Workforce for the Information Economy," 2001.

[3] A. Q. Gates, C. Casas, C. Servin, and M. Slattery, "Using Peer-Led Team Learning to Build University-Community College Relationships," in *Frontiers in Education Conference (FIE)*, 2015, pp. 1–7.

[4] J. Margolis, J. Goode, and K. R. Binning, "Exploring Computer Science: Active Learning for Broadening Participation in Computing," *Computing Research News*, Oct-2015.

[5] American Society for Engineering Education, "Going the Distance: Best Practices and Strategies for Retaining Engineering, Engineering Technology and Computing Students," *American Society for Engineering Education Report*. American Society for Engineering Education, p. 32, 2012.

[6] R. G. Book, "Women Recruitment / Retention in STEM Fields," in *2011 Midwest Section Conference of the American Society for Engineering Education*, 2011.

[7] M. Guzdial, "Top 10 Myths About Teaching Computer Science," *BLOG@CACM*

(Communications of the ACM), 2015. [Online]. Available: https://cacm.acm.org/blogs/blog-cacm/189498-top-10-myths-about-teaching-computer-science/fulltext. [Accessed: 05-Feb-2018].

[8] L. H. Jamieson and J. R. Lohmann, "Innovation With Impact," in *ASEE Conference*, 2012, pp. 1–77.

[9] R. M. Marra, K. A. Rodgers, D. Shen, and B. Bogue, "Leaving Engineering : A Multi-Year Single Institution Study," *J. Eng. Educ.*, vol. 101, no. 1, pp. 6–27, 2012.

[10] National Center for Women and Information Technology, "How Does Engaging Curriculum Attract Students to Computing?," 2010.

[11] L. L. Werner, B. Hanks, and C. McDowell, "Pair-programming helps female computer science students," *J. Educ. Resour. Comput.*, vol. 4, no. 1, 2004.

[12] L. Williams, E. Wiebe, K. Yang, M. Ferzli, and C. Miller, "In Support of Pair Programming in the Introductory Computer Science Course," *Comput. Sci. Educ.*, vol. 12, no. 3, pp. 197–212, 2002.

[13] S. Horwitz and S. H. Rodger, "Using peer-led team learning to increase participation and success of under-represented groups in introductory computer science," *ACM SIGCSE Bull.*, vol. 41, no. 1, pp. 163–167, 2009.

[14] J. P. Cohoon, "An introductory course format for promoting diversity and retention," *ACM SIGCSE Bull.*, vol. 39, no. 1, p. 395, 2007.

[15] J. M. Griffin, "Learning by Taking Apart," in *Proceedings of the 17th Annual Conference on Information Technology Education - SIGITE '16*, 2016, pp. 148–153.

[16] C. S. Dweck, *Mindset*, vol. 1, no. 1. Ballantine Books, 2010.

[17] J. M. Graham and R. Caso, "Measuring engineering freshman attitudes and perceptions of their first year academic experience: the continuing development of two assessment instruments," in *32nd Annual Frontiers in Education*, vol. 2, p. F3B–6–F3B–11.

[18] A. Hoegh and B. M. Moskal, "Examining science and engineering students' attitudes toward computer science," in *2009 39th IEEE Frontiers in Education Conference*, 2009, pp. 1–6.

[19] T. A. Lamb, "An imperfect but useful tool: The retrospective pretest," *Eval. Exch.*, vol. 11, no. 2, 2005.

[20] M. Hertz, "What do CS1 and CS2 mean? Investigating differences in the early courses," *41st ACM Tech. Symp. Comput. Sci. Educ.*, pp. 199–203, 2010.

[21] R. Johns, "One Size Doesn't Fit All: Selecting Response Scales For Attitude Items," *J. Elections, Public Opin. Parties*, vol. 15, no. 2, pp. 237–264, Sep. 2005.

[22] J. A. Krosnick *et al.*, "The Impact of 'No Opinion' Response Options on Data Quality: Non-Attitude Reduction or an Invitation to Satisfice?," *The Public Opinion Quarterly*, vol. 66. Oxford University PressAmerican Association for Public Opinion Research, pp. 371–403, 2002.

Appendix A

Student Reflections from Focus Group Discussions

- ” … in person, I can usually just raise my hand and someone will come over and help me. When you're online, it's easy to get out of it online, if you don't do the reading or we don't watch the videos, that's our choice. But, we go in every week to 291 (CTL support class) and get actual hands-on experience that we're missing out from the lecture.”

- “…it was nice working out these concepts with other people because I made friends that way, and I got to see we’re also on the same, everyone else is still just as clueless as I was about some basic things.”

- “I think that it just helps so much to be able to talk through with another person… and then ask questions because I feel like that's so different than just being taught something in lecture, and then, having to do it on your own because a lot of times it's like if I'm stuck, then I'm just stuck. I can look at stuff online, but sometimes that doesn't help nearly as much as talking through the problem with someone and working in teams.”

- “Especially coming from having no experience, it's helped a lot and put me on the same level as everyone else. Even the aspects of having other students to work with and the exams, alone, are worth it to most kids. I feel like we've all felt more comfortable taking the exams and that's obviously helpful for anyone in the class, that's a big part of it, too.”

- “I think CTL really taught us how to teach each other concepts because of the nature of the activities that we did.”

- "You kinda say to yourself like, ‘Oh, I can do programming,’ but you don't really feel super proficient in it yet. Until, after about a year of doing programming, then I felt really confident afterwards but, I felt like the CTL started a really good foundation for that as well, and it complimented the 231 (Intro) course.”

- “Assuming that 291 (CTL) is not required, I think there should at least be some computational thinking taught as more of a focus of 231 (Intro) because, from my own memory and then also what I'm observing with this sophomore I'm currently helping, it seems like it's just a lot of weekly projects, crank out the code, get it done, blah, blah, blah, which to improve your coding skill, obviously you have to code but I think there's also such importance in computational thinking ... I think there needs to be more of a balance is what I'm trying to say, between not just cranking out lines of code, but also, let's really think about it and understand what we're doing and what we're writing.”

- “I think it not only teaches us all the things that we mentioned but it also taught us like how to debug code, which I think is extremely, extremely important. It's so important that you would not be able to get nearly as far without it, obviously, but just the way that we stepped through line-by-line and really broke down code. I feel like if you don't know how to debug then it's almost impossible to figure out what's wrong with your program or what your program’s doing.”